DETAILED ACTION

1.   Claims 1-40 are pending in this action with claims 1, 30 newly amended, Claim
40 newly presented; and claims 15-29 are withdrawn.

Response to Arguments

2.   **Applicant's arguments filed 27 Oct 09 regarding objection to 37 CFR 1.75(c)
have been fully considered but they are not persuasive. However, upon consultation
with 101 panel, the objection has been withdrawn on other grounds.**

Briefly, discussion on differences between a method of using zero and an apparatus claim
using a zero were discussed, as well as the implication of *In re Gulack* and *In re Ngai.*

3.   **Applicant's arguments with respect to claims 1, 30 have been considered but
are moot in view of the new ground(s) of rejection.**

For interpretation of negative limitation, please note the examiner-initiated interview
summary accompanying this action.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all
obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

5.   **Claims 1-3, 5, 8, 9, 11, 14, 30-35, 40 are rejected under 35 U.S.C. 103(a) as**

obvious over Vahalia (US 6,389,420 B1) in view of Yamakawa et al. (US

2006/0069665 A1) hereinafter Yamakawa

With respect to claim 1, Vahalia discloses A method for attempting to access a first data

entity in a file system (Abstract, access to file data), the method being performed by a

first computing entity (Fig. 2, clients), the file system also including one or more

additional data entities that are concurrently accessible to at least one other computing

entity (Fig. 2, file systems accessible by multiple clients), the method comprising:

the first computing entity attempting to obtain a lease for itself on the first data entity

without needing another computing entity acting on its behalf (Fig. 2, clients do not

require other clients to act upon their behalf), by reading, using a processor (Col 3 lines

18, 35, all components use processors), an owner field (Fig. 5, ClientID) […] that can be

used to determine whether the first data entity is leased by a computing entity (Col 15

lines 35-41, owner is used to determine) and a time field […] (Fig. 5, Grant

time/Expiration time) that can be used to determine whether a lease for the first data

entity has expired (Col 8 lines 1-2, file manager timer in RAM) and:

if the owner field indicates that the first data entity is not currently leased, the first

computing entity writing to the owner field in the file system to indicate an assumption of

a lease of the first data entity (Fig. 8 element 88, records lock grant. Also see Fig. 5,

Owner ID field) and writing to the time field in the file system to indicate when the lease

expires (Col 8 lines 1-8, lock manager stores timer in a linked list); else

if the owner field indicates that the first data entity has been leased (Fig. 8 element 86,

incompatible lock exists), the first computing entity reading the time field in the file

system and:

> if the time field indicates that the lease has expired (Fig. 8, compare
>
> element 86 and 92; expired locks allow other clients to access. Also, note
>
> that this corresponds a move to "D" from element 86), the first computing
>
> entity writing to the owner field in the file system to break the existing
>
> lease and to indicate an assumption of a new lease (Fig. 8 element 88,
>
> Lock grant recorded) and the first computing entity writing to the time
>
> field in the file system to indicate when the new lease expires (Fig. 8
>
> element 88, expiration time recorded); else

> if the time field indicates that the lease is still active (Fig. 8 element 86,
>
> YES), concluding that the first data entity is currently unavailable (Fig. 8
>
> element 90, client is waitlisted); and if a lease is obtained, the first
>
> computing entity accessing the first data entity while the lease is in effect
>
> (Col 11 lines 23-27, describing during lock renewal requests: if a client is
>
> waitlisted then the previous lease won't be renewed and presumably the
>
> first-priority waitlisted client will get the lease).

Vahalia does not appear to explicitly state that the owner fields and time fields are in the

file system. In fact, the cited portions for said information storage indicate that they are

handled by the file manager. However, each file managers in one embodiment (See

embodiment shown in Fig. 2, compare with Fig. 1) is coupled to an individual file system

on a cached disk array. Furthermore, note that file locking information and metadata

were described in relation to Fig. 1 as being stored on the cached disk array, and not

stored on a file manager. Fig. 2 is discussed in reference to Fig. 1 as an alternate

embodiment, but only management is discussed, so one skilled in the art would

understand that the implementation of storage is part of the file system of the cached disk

array. Thus, one skilled in the art would understand that that the file manager disclosed

in the reference is, in fact, part of the file system, and that the designations "file manager"

and "file system" are lexicographically created in the Vahalia reference to differentiate

organized data and the access functions of said data.

To the extent that the file manager is arguably not part of the file system, Yamakawa

discloses that said functions were known to reside in a file system ([0150] file system

info includes grace-time and user identifiers), and that their separation into separate

object classes is mere design choice. It would have been obvious to one skilled in the art

at the time of invention to combine the teaching of the references for 1) as a personal

preference of design choice 2) ease of programming such that setting permissions file

system objects such that the file manager objects instantiated from file manager classes

have free access to their respectively coupled file system.

With respect to claim 2, Vahalia discloses the first data entity is a file (Abstract, clients may share access to a file).

With respect to claim 3, Vahalia discloses the first data entity includes metadata (Col 19 lines 14-17, inodes are one type of "preferred metadata") and the owner field is located in this metadata (Col 20 line 66-Col 21 line 9, inodes are changed to reflect ownership change. Therefore, inodes must contain an owner field).

With respect to claim 5, Vahalia discloses the step of the first computing entity writing to the owner field to indicate an assumption of a lease of the first data entity comprises the first computing entity writing a data value to the owner field (Col 8 lines 47-50, "particular clients") that uniquely identifies the first computing entity (Col 8 lines 50-51, "client identification field").

With respect to claim 8, Vahalia discloses a lease expires a predetermined period of time after the lease begins (Col 7 lines 37-45, predetermined time is delta sub 2), and wherein the step of the first computing entity writing to the time field to indicate when the lease expires comprises the first computing entity writing a current time value to the time field (Col 8 lines 48-58, one time field corresponds to expiration time).

With respect to claim 9, Vahalia discloses the first computing entity determines whether a prior lease has expired by reading a first value from the time field (Fig. 5 Lock time), delaying for a predetermined lease period (Fig. 3, delta sub 1)

and reading a second value from the time field (Fig. 5, Grant time), wherein the first

computing entity determines that the prior lease has expired if the second value is the

same as the first value, and the first computing entity determines that the prior lease has

not expired if the second value is different from the first value (Col 7 lines 54-59, grace

period defined as a operation of t1, t0, delta sub 2 and delta sub 1. If atomic differences

in time are not breached, then deltas are the same and ts are the same, thus no grace

period is required (and time must therefore be expired)).

With respect to claim 11, Vahalia discloses if the first computing entity concludes that

the first data entity is currently unavailable (Fig. 5 elements 85, 86, 90, not a renewal so a

new request, but an existing compatible lock exists), the first computing entity further

writes an entry to a queue owner field in a queue in the file system to indicate an interest

in accessing the first data entity (Fig. 5 element 90, client is put on a wait list).

With respect to claim 14, Vahalia discloses if a lease is obtained, the first computing

entity also sets a renewal timer and, after the renewal timer expires, the first computing

entity renews the lease by writing a new value to the time field (Fig. 8 element 88, if

triggered through the paths).

With respect to claim 30, Vahalia discloses A method for attempting to access a first data

entity in a file system (Abstract, access to file data), the method being performed by a

first computing entity (Fig. 2, clients), the file system also including one or more

additional data entities that are concurrently accessible to at least one other computing

entity (Fig. 2, file systems accessible by multiple clients), the method comprising:

the first computing entity attempting to access the first data entity without needing

another computing entity acting on its behalf (Fig. 2, clients do not require other clients

to act upon their behalf), reading, using a processor (Col 3 lines 18, 35, all components

use processors), an owner field (Fig. 5, ClientID) […] that can be used to determine

whether the first data entity is in use by a computing entity (Col 15 lines 35-41, owner is

used to determine) and determining whether the first data entity is in use by a computing

entity;

if the first data entity is not in use by a computing entity, the first computing entity

writing to the owner field in the file system to take control of a lock on the first data

entity (Fig. 8 elements 86-89, if no existing incompatible lock, then the lock can be

granted to the entity); and

if control of the lock is obtained, the first computing entity accessing the first data entity

(Id.); else

if control of the lock is not obtained (Fig. 5 elements 85, 86, 90, not a renewal so a new

request, but an existing compatible lock exists), the first computing entity writing an

entry to a queue owner field in the file system to indicate an interest in accessing the first

data entity and waiting for an opportunity to access the first data entity (Fig. 5 element

90, client is put on a wait list).

Vahalia does not appear to explicitly state that the owner fields and time fields are in the
file system. In fact, the cited portions for said information storage indicates that they are
handled by the file manager. However, each file managers in one embodiment (See
embodiment shown in Fig. 2, compare with Fig. 1) is coupled to an individual file system
on a cached disk array. Furthermore, note that file locking information and metadata
were described in relation to Fig. 1 as being stored on the cached disk array, and not
stored on a file manager. Fig. 2 is discussed in reference to Fig. 1 as an alternate
embodiment, but only management is discussed, so one skilled in the art would
understand that the implementation of storage is part of the file system of the cached disk
array. Thus, one skilled in the art would understand that that the file manager disclosed
in the reference is, in fact, part of the file system, and that the designations "file manager"
and "file system" are lexicographically created in the Vahalia reference to differentiate
organized data and the access functions of said data.

To the extent that the file manager is arguably not part of the file system, Yamakawa
discloses that said functions were known to reside in a file system ([0150] file system
info includes grace-time and user identifiers), and that their separation into separate
object classes is mere design choice. It would have been obvious to one skilled in the art
at the time of invention to combine the teaching of the references for 1) as a personal
preference of design choice 2) ease of programming such that setting permissions file
system objects such that the file manager objects instantiated from file manager classes
have free access to their respectively coupled file system.

With respect to claim 31, Vahalia discloses if the first data entity is in use by a computing

entity, the first computing entity reading a time field in the file system to determine

whether a lease on the data entity has expired and, if the lease has expired (Fig. 8,

compare element 86 and 92; expired locks allow other clients to access. Also, note that

this corresponds a move to "D" from element 86), the first computing entity writing to the

owner field to break the existing lease (Fig. 8 element 88, Lock grant recorded) and to

indicate an assumption of a new lease of the first data entity (Fig. 8 elements 88, 89.

Actual lock grant proceeds the recording step).

With respect to claim 32, Vahalia discloses the first computing

entity determines whether a prior lease has expired by reading a first value from the time

field (Fig. 5 Lock time), delaying for a predetermined lease period (Fig. 3, delta sub 1)

and reading a second value from the time field (Fig. 5, Grant time), wherein the first

computing entity determines that the prior lease has expired if the second value is the

same as the first value, and the first computing entity determines that the prior lease has

not expired if the second value is different from the first value (Col 7 lines 54-59, grace

period defined as a operation of t1, t0, delta sub 2 and delta sub 1. If atomic differences

in time are not breached, then deltas are the same and ts are the same, thus no grace

period is required (and time must therefore be expired)).

With respect to claim 33, Vahalia discloses if the first data entity is not in use by a

computing entity, in addition to writing to the owner field to take control of the lock on

the first data entity (Fig. 8 element 88, records grant), the first computing entity writing to

a time field in the file system to indicate when a lease of the first data entity expires (Fig.

8 element 88).

With respect to claim 34, Vahalia discloses the first data entity is a file (Abstract, clients

may share access to a file).

With respect to claim 35, Vahalia discloses the first data entity

includes metadata (Col 19 lines 14-17, inodes are one type of "preferred metadata") and

the owner field is located in this metadata (Col 20 line 66-Col 21 line 9, inodes are

changed to reflect ownership change. Therefore, inodes must contain an owner field).

With respect to claim 40, Vahalia discloses the owner field is stored in a file system lock

structure (Col 16 lines 13-23, various elements of structures that store lists of owners.) in

a *file manager*

Vahalia does not appear to explicitly state that the owner fields and time fields are in the

file system. In fact, the cited portions for said information storage indicates that they are

handled by the file manager. However, each file managers in one embodiment (See

embodiment shown in Fig. 2, compare with Fig. 1) is coupled to an individual file system

on a cached disk array. Furthermore, note that file locking information and metadata

were described in relation to Fig. 1 as being stored on the cached disk array, and not

stored on a file manager. Fig. 2 is discussed in reference to Fig. 1 as an alternate

embodiment, but only management is discussed, so one skilled in the art would

understand that the implementation of storage is part of the file system of the cached disk

array. Thus, one skilled in the art would understand that that the file manager disclosed

in the reference is, in fact, part of the file system, and that the designations "file manager"

and "file system" are lexicographically created in the Vahalia reference to differentiate

organized data and the access functions of said data.

To the extent that the file manager is arguably not part of the file system, Yamakawa

discloses that said functions were known to reside in a file system ([0150] file system

info includes grace-time and user identifiers), and that their separation into separate

object classes is mere design choice. It would have been obvious to one skilled in the art

at the time of invention to combine the teaching of the references for 1) as a personal

preference of design choice 2) ease of programming such that setting permissions file

system objects such that the file manager objects instantiated from file manager classes

have free access to their respectively coupled file system.

6.   **Claims 4, 36 are rejected under 35 U.S.C. 103(a) as obvious over Vahalia**

with clarification by The Authoritative Dictionary of IEEE Standards Terms, 7th Ed

(ISBN: 0-7381-2601-2) hereinafter The Dictionary as applied to claims 1, 30, in view

of Yamakawa.

With respect to claim 4, Vahalia discloses a first data entity is a file (Abstract).

The Dictionary states under it's entry for directory (5) a *file* that contains directory

entries. No two entries in a directory shall have the same filename. (emphasis added)

Because *directory* is a subset of *file* (namely, a file that contains files), and Vahalia

discloses locking and granting access to files, Vahalia discloses a first data entity is a

directory.

With respect to claim 36, Vahalia discloses a first data entity is a file (Abstract).

The Dictionary states under it's entry for directory (5) a *file* that contains directory

entries. No two entries in a directory shall have the same filename. (emphasis added)

Because *directory* is a subset of *file* (namely, a file that contains files), and Vahalia

discloses locking and granting access to files, Vahalia discloses a first data entity is a

directory.

Claim Rejections - 35 USC § 103

7.   **Claims 6, 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over**

Vahalia in view of Yamakawa as applied to claims 1, 30, in view of Windows 2000

Support, How to specify an IP address (http://support.microsoft.com/kb/308199)

hereinafter W2kIP.

With respect to claim 6, Vahalia does not directly disclose the data value that uniquely

identifies the first computing entity is determined autonomously by the first computing

entity.

W2kIP discloses a method of entering a data value that uniquely identifies a first

computing entity (How to specify an IP Address, steps 2-3). IP Addresses are well

known to uniquely identify a computer over a network. Thus, W2kIP discloses the data

value that uniquely identifies the first computing entity (IP address entry) is determined

autonomously by the first computing entity (no DNS access required; all performed on

the computer via control panel.).

Vahalia is directed to serving clients and W2kIP discloses how to put computers

(including clients) onto a network. It would have been obvious to one skilled in the art at

the time of invention to combine the teachings of Vahalia and W2kIP in order to allow

clients to access a network.

With respect to claim 38, Vahalia discloses the first computing entity assumes a lock on

the first data entity by writing the unique data value into the owner field (Col 8 lines 50-

51, "client identification field"; Co 8 lines 47-50, said clients are "particular clients").

Vahalia does not directly disclose the data value that uniquely identifies the first

computing entity is determined autonomously by the first computing entity.

W2kIP discloses a method of entering a data value that uniquely identifies a first

computing entity (How to specify an IP Address, steps 2-3). IP Addresses are well

known to uniquely identify a computer over a network. Thus, W2kIP discloses the data

value that uniquely identifies the first computing entity (IP address entry) is determined

autonomously by the first computing entity (no DNS access required; all performed on the computer via control panel.).

Vahalia is directed to serving clients and W2kIP discloses how to put computers (including clients) onto a network. It would have been obvious to one skilled in the art at the time of invention to combine the teachings of Vahalia and W2kIP in order to allow clients to access a network without requiring a separate server to regulate client identity.

8.   **Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Vahalia in view of Yamakawa as applied to claim 1, in view of Official Notice.**
With respect to claim 7, Vahalia discloses special characters used for invalid entries (Col 8 line 67 - Col 9 line 1) but does not disclose the owner field indicates that the first data entity is not currently leased when the owner field contains a value of zero.
Official Notice is taken that it was well known at the time of intention to 1) provide sentinel values 2) that 'zero' is one of two fundamental digits of operation on a computer (the alternative being 'one') and 3) cardinality of states is a precondition of computability.

Vahalia and Official Notice fall within the field of computing. It would have been obvious to one skilled in the art at the time of invention to severally combine the teachings of Vahalia with Official Notice(1) in order to indicate an available data entity without instantiating a new data structure; Vahalia with Official Notice (2) as a mere

design choice; Vahalia with Official Notice(3) in order to indicate that a data entity is available.

9.   **Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Vahalia in view of Yamakawa as applied to claim 1, in view of Shaughnessy (US 5,692,178) hereinafter Shaughnessy.**

Vahalia does not directly disclose the steps of reading the owner field and reading the time field are both performed in a single read operation.

Shaughnessy discloses a file lock read in a single I/O operation (Col 19 lines 15-16). Both Vahalia and Shaughnessy are directed to locking. It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teaching of Vahalia and Taylor to include the single I/O operation of Shaughnessy for reading the owner field and reading the time field are both performed in a single read operation because this avoids multiple reads, thus avoiding a performance penalty (Shaughnessy, Col 19, lines 24-26).

10.   **Claims 12, 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vahalia in view of Yamakawa as applied to claims 1, 30, in view of Taylor (US 7,107,267 B2) hereinafter Taylor.**

With respect to claim 12, Vahalia does not appear to directly disclose the first computing entity also writes to a queue time field in the queue in the file system to indicate a period of time for which the entry to the queue owner field is valid.

Taylor discloses a lease length (Col 5 line 57; Fig. 3 element 118) as part of the entry of a
queue.

Both Vahalia and Taylor are directed towards locking mechanisms, with the base
mechanism including a queue. One of the queues in Taylor was improved with a time
field for expiration. It would have been obvious to one of ordinary skill in the art at the
time of applicant's invention to improve the Vahalia queue to include the lease length of
Taylor so that a process granted access would cease access operations when the lease
expires (Taylor, column 5, lines 44-45), in order to obtain the predictable result of
removal from the queue.

With respect to claim 39, Vahalia discloses if control of the lock is not obtained (Fig. 5
elements 85, 86, 90, not a renewal so a new request, but an existing compatible lock
exists), in addition to writing an entry to a queue owner field in the file system to indicate
an interest in accessing the first data entity (Fig. 5 element 90, client is put on a wait list),
the first computing entity writing to a queue time field in the file system to indicate a
period of time for which the entry to the queue owner field is valid.

Taylor discloses a lease length (Col 5 line 57; Fig. 3 element 118) as part of the entry of a
queue.

Both Vahalia and Taylor are directed towards locking mechanisms, with the base
mechanism including a queue. One of the queues in Taylor was improved with a time
field for expiration. It would have been obvious to one of ordinary skill in the art at the
time of applicant's invention to improve the Vahalia queue to include the lease length of
Taylor so that a process granted access would cease access operations when the lease

expires (Taylor, column 5, lines 44-45), in order to obtain the predictable result of
removal from the queue.

11.   **Claims 13, 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over
Vahalia in view of Yamakawa as applied to claims 1, 30, in view of Stakutis et al.
(US 6,658,417) hereinafter Stakutis.**

With respect to claim 13, Vahalia does not specifically disclose the first computing entity
reserving a disk on which the owner field and the time field are located to ensure
exclusive access to the disk for the reading and writing of the owner field and the time
field.

Stakutis discloses reserving a disk on which the owner field and the time field are located
to ensure exclusive access to the disk for the reading and writing of the owner field and
the time field (Appendix A Section 3.6.1 allows locking of a file. Col 7 lines 17-19, a
"file" is generic name for any arbitrary storage structure.).

Both Vahalia and Stakutis are directed to shared storage. It would have been obvious to
one of ordinary skill in the art at the time of applicant's invention to modify the teaching
to allow the client nodes to access the file system without extraneous network
communications (Stakutis, column 4, lines 27-29).

With respect to claim 37, Vahalia does not specifically disclose the first computing entity
reserving a disk on which the owner field and the time field are located to ensure

exclusive access to the disk for the reading and writing of the owner field and the time field.

Stakutis discloses reserving a disk on which the owner field and the time field are located to ensure exclusive access to the disk for the reading and writing of the owner field and the time field (Appendix A Section 3.6.1 allows locking of a file. Col 7 lines 17-19, a "file" is generic name for any arbitrary storage structure.).

Both Vahalia and Stakutis are directed to shared storage. It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching to allow the client nodes to access the file system without extraneous network communications (Stakutis, column 4, lines 27-29).

Remarks

The examiner notes that Vahalia Fig. 5 describes a log. However, (1) the broad interpretation of storing in fields encompasses said logs as an explicit disclosure and (2) even if the claims were further limited to be restricted to data structures, the information being stored in the logs must necessarily reflect information that is contained in data structures and thus Vahalia might also provide implicit disclosure.

Conclusion

12.   **The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.**

http://www.windowsnetworking.com/articles_tutorials/w2ktcpip.html

> Identical interface for W2K network adaptor IP assignment shows that IP
>
> assignment functionality existed in 2001.

13.   Applicant's amendment necessitated the new ground(s) of rejection presented in
this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP
§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37
CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE
MONTHS from the mailing date of this action. In the event a first reply is filed within
TWO MONTHS of the mailing date of this final action and the advisory action is not
mailed until after the end of the THREE-MONTH shortened statutory period, then the
shortened statutory period will expire on the date the advisory action is mailed, and any
extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the
advisory action. In no event, however, will the statutory period for reply expire later than
SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the
examiner should be directed to JASON LIAO whose telephone number is (571)270-3775.
The examiner can normally be reached on M-Th.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's
supervisor, Pierre Vital can be reached on 571-272-4215. The fax phone number for the
organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status

information for unpublished applications is available through Private PAIR only. For

more information about the PAIR system, see http://pair-direct.uspto.gov. Should you

have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO

Customer Service Representative or access to the automated information system, call

800-786-9199 (IN USA OR CANADA) or 571-272-1000.

11 Jan 10
/Jason Liao/
Examiner, Art Unit 2156

/Pierre M. Vital/
Supervisory Patent Examiner, Art Unit 2156